Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A method of optimizing the operation of ~~operating~~ a processing element ~~to calculate a local extrema, the method~~ comprising:

serially loading, on each clock cycle, values in odd positions within a set of values into a first register and values in even positions within said set into a second register;

determining a series of first extrema from said odd set when values in even positions are being loaded and saving a current value for said first extrema in a third register when a value in an odd position is being loaded;

determining a series of second extrema from said even set when values in odd positions are being loaded and saving a current value for said second extrema in a fourth register when a value in an even position is being loaded;

determining a local extrema from values in said third and fourth registers; and

storing said local extrema.

2. (previously presented) The method of claim 1 wherein said serial loading said set of values comprises:

loading a value from an odd position within said set into said first register;

loading a value from an even position within said set into said second register and transferring said value in said first register to said third register;

loading a value from a next odd position within said set into said first register and transferring said value in said second register to said fourth register; and

loading a value from a next even position within said set into said second register.

3. (previously presented) The method of claim 2 wherein said determining a first series of extrema from said odd set comprises comparing the value in said first register to the value in said third register.

4. (previously presented) The method of claim 3 wherein said determining further comprises:

selecting the greater value from said first register and said third register if a high first extrema is desired; and

selecting the lesser value from said first register and said third register if a low first extrema is desired.

5. (previously presented) The method of claim 2 wherein said determining a second series of extrema from said even set of values comprises comparing the value in said second register to the value in said fourth register.

6. (previously presented) The method of claim 5 wherein said determining further comprises:

selecting the greater value from said second register and said fourth register if a high second extrema is desired; and

selecting the lesser value from said second register and said fourth register if a low second extrema is desired.

7. (previously presented) The method of claim 1 wherein said determining said local extrema further comprises:

selecting the greater value from said third and fourth registers if a local high extrema is desired; and

selecting the lesser value from said third and fourth registers if a local low extrema is desired.

8. (previously presented) The method of claim 2 further comprising:

storing a first in said series of first extrema in said third register;

loading another value from an odd position within said set into said first register;

comparing the value in said first register to the value in said third register; and

repeating said storing, loading and comparing steps for remaining values within an odd position within said set.

9. (previously presented) The method of claim 2 further comprising:

storing a first in said series of second extrema in said fourth register;

loading another value from an even position within said set into said second register;

comparing the value in said second register to the value in said fourth register; and

repeating said storing, loading and comparing steps for remaining values within an even position within said set.

10. cancelled.

11. cancelled.

12. (previously presented) The method of claim 1 wherein said determining a series of first extrema from said odd set comprises comparing a least significant byte of a short value from said odd position within said set to a least significant byte of a short value from another odd position within said set and comparing a most significant byte of said short value from said odd position within said set to a most significant byte of said short value from said another odd position within said set.

13. (previously presented) The method of claim 1 wherein said determining a series of second extrema from said even set comprises comparing a least significant byte of a short value from said even position within said set to a least significant byte of a short value from another even position within said set and comparing a most significant byte of said short value from said even position within said set to a most significant byte of said short value from said another even position within said set.

14. – 31. cancelled.

32. (currently amended) A processing element, comprising:

an arithmetic logic unit;

condition logic responsive to said arithmetic logic unit;

a plurality of registers connected to a bus and responsive to said arithmetic logic unit;

a result pipeline responsive to said arithmetic logic unit;

an interface; and

register files connected between said interface and said result pipeline; said processing element configured to:

serially load, on each clock cycle, values in odd positions within a set of values into a first register and values in even positions within said set into a second register;

said arithmetic logic unit determining a series of first extrema from said odd set <u>when values in even positions are being loaded</u> and saving a current value for said first extrema in a third register <u>when a value in an odd position is being loaded</u>;

said arithmetic logic unit determining a series of second extrema from said even set when values in odd positions are being loaded and saving a current value for said second extrema in a fourth register when a value in an even position is being loaded;

said arithmetic logic unit determining a local extrema from values in said third and fourth registers; and

output said local extrema through said result pipeline and said interface.

33. (previously presented) The processing element of claim 32 wherein said arithmetic logic unit is configured to determine a series of first extrema from said odd set by comparing the value in said first register to the value in said third register.

34. (previously presented) The processing element of claim 33 wherein said arithmetic logic unit is configured to:

select the greater value from said first register and said third register if a high first extrema is desired; and

select the lesser value from said first register and said third register if a low first extrema is desired.

35. (previously presented) The processing element of claim 32 wherein said arithmetic logic unit is configured to determining a second series of extrema from said even set of values by comparing the value in said second register to the value in said fourth register.

36. (previously presented) The processing element of claim 35 wherein said arithmetic logic unit is configured to:

select the greater value from said second register and said fourth register if a high second extrema is desired; and

select the lesser value from said second register and said fourth register if a low second extrema is desired.

37. (previously presented) The processing element of claim 32 wherein said arithmetic logic unit is configured to:

select the greater value from said third and fourth registers if a local high extrema is desired; and

select the lesser value from said third and fourth registers if a local low extrema is desired.

38. (previously presented) The processing element of claim 32 wherein the processing element is configured to:

serially load another value from an odd position within said set into said first register;

compare the value in said first register to the value in said third register; and

repeat said storing, loading and comparing steps for remaining values within odd positions within said set.

39. (previously presented) The processing element of claim 32 wherein said processing element is configured to:

serially load another value from an even position within said set into said second register;

comparing the value in said second register to the value in said fourth register; and

repeat said storing, loading and comparing steps for remaining values within even positions within said set.